

LISTING OF THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A system comprising:
a first node that provides a broadcast request for data, the first node including a conflict state machine for managing non-data responses to the broadcast request for the data provided from the first node, the first node receiving a read conflict response to the broadcast request from the first node, the read conflict response indicating that a second node has a pending broadcast read request for the data, the conflict state machine transitioning to a conflict state in response to the first node receiving the read conflict response; and
a third node that provides the requested data to the first node in response to the broadcast request from the first node, the first node filling the data provided by the third node in a cache associated with the first node based on the state of the conflict state machine.
2. (Original) The system of claim 1, wherein the broadcast request provided by the first node is a source broadcast read request.
3. (Original) The system of claim 2, wherein the first node provides a read conflict response to the broadcast read request from the second node, the read conflict response provided by the first node indicating that the broadcast read request of the first node conflicts with the pending broadcast read request of the second node.
4. (Original) The system of claim 3, wherein the third node provides the requested data to the second node in response to the broadcast read request from the second node, the second node filling the data provided by the third node in a cache associated with the second node.
5. (Original) The system of claim 1, wherein the request for data broadcast by the first node is a source broadcast write request.
6. (Original) The system of claim 5, wherein the first node provides a second conflict response to the pending broadcast read request from the second node, the second conflict

response provided by the first node indicating that the write request broadcast by the first node conflicts with the broadcast read request from the second node.

7. (Currently Amended) The system of claim 6, wherein the broadcast request provided by the first node is broadcast using a first cache coherency protocol, the first cache coherency protocol being chosen by the first node based on the state of the conflict state machine, the second node in response to the second conflict response provided by the first node reissues the pending broadcast read request of the second node.

8. (Original) The system of claim 7, wherein the first cache coherency protocol is a source broadcast cache coherency protocol and the second node reissues the broadcast read request using a forward progress cache coherency protocol.

9. (Original) The system of claim 6, wherein the third node provides the requested data to the second node in response to the pending broadcast read request of the second node, the second conflict response provided by the first node preventing the second node from filling the data provided by the third node in a cache associated with the second node.

10. (Original) The system of claim 6, wherein the third node provides shared data to the second node in response to the pending broadcast read request of the second node, the second node filling a cache associated with the second node with the shared data and associating an invalid state with the shared data filled in the cache associated with the second node.

11. (Original) The system of claim 1, wherein the third node comprises one of a home node and an owner node.

12. (Currently Amended) The system of claim 1, wherein the broadcast request provided by the first node is broadcast using a source broadcast cache coherency protocol, the source broadcast cache coherency protocol being chosen by the first node based on the state of the conflict state machine.

13. (Original) The system of claim 1, wherein the first node defines a first processor and the second node defines a second processor, the first and second processors having an associated

cache, the associated caches of the first and second processors each comprising a plurality of cache lines, each cache line having a respective tag address that identifies associated data and each cache line having state information that indicates a state of the associated data for the respective cache line, the first and second processors being capable of communicating with each other and with other nodes of the system through an interconnect.

14. (Original) The system of claim 13, further comprising a first cache controller associated with the first processor and a second cache controller associated with the second processor, the first cache controller being operative to manage data requests and responses for the associated cache of the first processor, the first cache controller effecting state transitions associated with the data in the associated cache of the first processor based on the data requests and responses for the associated cache of the first processor, the second cache controller being operative to manage data requests and responses for the associated cache of the second processor, the second cache controller effecting state transitions associated with the data in the associated cache of the second processor based on the data requests and responses for the associated cache of the second processor.

15. (Original) The system of claim 13, wherein the system implements a hybrid cache coherency protocol wherein each of the first, second, and third processors employs a source broadcast-based protocol to issue a request for the data and provide responses for the data, and employs an associated second protocol to reissue a request for the data in response to the request failing in the source broadcast protocol, the second protocol employing a forward progress technique.

16. (Currently Amended) A multi-processor network comprising:

a first processor node operative to issue a first source broadcast request for data, the first processor node including an associated conflict state machine for managing non-data responses to the first source broadcast request for the data;

a second processor node operative to issue a second source broadcast request for the data;
and

a third node operative to provide a data response in response to the respective source broadcast requests of the first and second processor nodes, the third node being one of an owner processor node and a memory node;

the second processor node being operative to provide a read conflict response to the first source broadcast request when the second source broadcast request is a read request, the second processor node being operative to provide a second conflict response to the first source broadcast request when the second source broadcast request is a write request;

the conflict state machine transitioning to a first conflict state of a plurality of conflict states in response to the first processor node receiving the read conflict response, and the conflict state machine transitioning to a second conflict state of the plurality of conflict states in response to the first processor node receiving the second conflict response;

the first processor node being operative ~~in response to receiving a read conflict response from the second processor~~ to implement a cache fill with the data provided by the third node if the conflict state machine transitions to the first conflict state.

17. (Currently Amended) The multi-processor network of claim 16, wherein the first processor node is operative ~~in response to a write conflict response from the second processor node~~ to issue a request for the data using a forward progress technique if the conflict state machine transitions to the second conflict state in response to the first processor node receiving the second conflict response.

18. (Currently Amended) The multi-processor network of claim 17, wherein ~~the second conflict response from the second processor node prevents~~ the first processor node is prevented from implementing the cache fill with the data provided by the third node if the conflict state machine transitions to the second conflict state in response to the first processor node receiving the second conflict response.

19. (Original) The multi-processor network of claim 16, wherein the first source broadcast request is one of a source broadcast write request and a source broadcast read request.

20. (Original) The multi-processor network of claim 16, wherein the source broadcast request issued by the first processor node exists concurrently with the source broadcast request issued by the second processor node.

21. (Currently Amended) The multi-processor network of claim 16, wherein the third ~~processor~~ node provides shared data to the second processor node in response to the second processor node providing the second source broadcast request as a broadcast read request, the second processor node filling the shared data in a cache associated with the second processor node and associating an invalid state with the data in the cache associated with the second processor node.

22. (Currently Amended) The multi-processor network of claim 16, wherein the third node is a third processor node and each of the first, second, and third processor nodes has an associated cache that comprises a plurality of cache lines, each cache line having a respective tag address that identifies associated data and having state information that indicates a state of the associated data for the respective cache line, the first, second, and third processor nodes being capable of communicating with each other and with other nodes of the system through an interconnect, the multi-processor network further comprising a first cache controller associated with the first processor node, a second cache controller associated with the second processor node, and a third cache controller associated with the third processor node, the first cache controller being operative to manage data requests and responses for the associated cache of the first processor, the first cache controller effecting state transitions associated with the data in the associated cache of the first processor based on the data requests and responses for the associated cache of the first processor, the second cache controller being operative to manage data requests and responses for the associated cache of the second processor, the second cache controller effecting state transitions associated with the data in the associated cache of the second processor based on the data requests and responses for the associated cache of the second processor, the third cache controller being operative to manage data requests and responses for the associated cache of the third processor, the third cache controller effecting state transitions associated with the data in the associated cache of the third processor based on the data requests and responses for the associated cache of the third processor.

23. (Currently Amended) The multi-processor network of claim 16, wherein the third node is third processor node the network implements a hybrid cache coherency protocol in which each of the first, second, and third processor nodes employs a source broadcast-based protocol to issue requests for data and provide responses to requests, and employs an associated protocol employing a forward progress technique to reissue a request for data in response to a request failing in the source broadcast protocol.

24. (Currently Amended) A computer system comprising:

a first processor operative to issue a source broadcast request for data the first processor including an associated conflict state machine for managing non-data responses to the first source broadcast request for the data;

a second processor operative to issue a source broadcast request for the data; and

a third processor operative to provide a data response to both the first and second processors in response to the source broadcast requests of the first and second processors;

the second processor in response to the source broadcast request of the first processor providing a read conflict response when the source broadcast request of the second processor is a source broadcast read request, the second processor in response to the source broadcast request of the first processor providing a second conflict response when the source broadcast request of the second processor is a source broadcast write request;

the conflict state machine transitioning to a first conflict state of a plurality of conflict states if the first processor receives the read conflict response, the conflict state machine transitioning to a second conflict state of the plurality of conflict states if the first processor receives the second conflict response having a priority higher than the priority of the first conflict response; and

the first processor ~~in response to the read conflict response of the second processor~~ being operative to fill the data provided by the third processor in a cache associated with the first processor if the conflict state machine transitions to the first conflict state in response to the first processor receiving the read conflict response.

25. (Currently Amended) The computer system of claim 24, wherein the first processor ~~in response to the second conflict response of the second processor~~ is operative to reissue the

source broadcast request from the first processor by issuing a request for the data employing a forward progress protocol if the conflict state machine transitions to the second conflict state in response to the first processor receiving the second conflict response.

26. (Currently Amended) The computer system of claim 24, wherein ~~the second conflict response from the second processor prevents~~ the first processor is prevented from filling the data provided by the third processor in the cache associated with the first processor if the conflict state machine transitions to the second conflict state in response to the first processor receiving the second conflict response.

27. (Original) The computer system of claim 24, wherein the third processor provides a shared data response to the first processor in response to the source broadcast request for the data, the first processor being operative to place the shared data in the cache associated with the first processor and associate an invalid state with the data in the cache associated with the first processor.

28. (Original) The computer system of claim 24, wherein the computer system implements hybrid cache coherency protocol in which each of the first, second, and third processor employs a source broadcast-based protocol to issue requests for data and provide responses to requests, and employs an associated protocol employing a forward progress technique to reissue a request for data in response to a request failing in the source broadcast protocol.

29. (Currently Amended) A system comprising:

means for providing a broadcast request for data from a first node using a first cache coherency protocol, the means for providing the broadcast request including means for managing non-data responses to the broadcast request and for transitioning among a plurality of conflict states in response to the non-data responses;

means for providing a read conflict response from a second node to the broadcast request from the first node when the second node has a pending broadcast read request for the data, wherein the means for managing non-data responses transitions to a conflict state of the plurality of conflict states according to a highest priority non-data response received by the means for providing the broadcast request;

means for providing the data from a third node to the first node in response to the broadcast request from the first node; and

means for placing the data from the third node in a cache associated with the first node in response to the read conflict response from the second node causing the means for managing non-data responses to transition to the conflict state.

30. (Original) The system of claim 29, wherein the means for providing a broadcast request from the first node comprises means for providing a broadcast read request for the data from the first node, the system further comprising:

means for providing a read conflict response from the first node to the second node in response to the broadcast read request of the second node, the read conflict response from the first node indicating that the pending broadcast read request of the second node conflicts with the broadcast read request for the data from the first node;

means for providing the requested data to the second node from the third node in response to the broadcast read request of the second node; and

means for filling the data provided to the second node by the third node in a cache associated with the second node in response to the second node receiving the read conflict response from the first node.

31. (Original) The system of claim 29, wherein the means providing a broadcast request from the first node comprises means for providing a broadcast write request for the data from the first node, the system further comprising:

means for providing a second conflict response from the first node to the second node in response to the pending broadcast read request of the second node, the second conflict response from the first node indicating that the pending broadcast read request of the second node conflicts with the broadcast write request for the data from the first node; and

means for reissuing the broadcast read request of the second node employing a forward progress protocol in response to the second conflict response provided by the first node.

32. (Original) The system of claim 31, further comprising means for preventing the second node from placing the data provided by the third node in a cache associated with the second node in response to the second conflict response provided by the first node.

33. (Currently Amended) The system of claim 29, wherein the means for providing the data from the third node to the first node is operative to provide shared data to the ~~first processor~~ means for providing a broadcast request, the means for placing the data provided by the third node placing the shared data in the cache associated with the first processor, the system further comprising means for associating an invalid state with the data in the cache associated with the first processor.

34. (Currently Amended) A method comprising:
providing a source broadcast request from a first node for data;
providing a read conflict response to the first node from a second node in response to the source broadcast request from the first node, the read conflict response indicating that the second node has a pending broadcast read request for the data;
transitioning a state of a conflict state machine associated with the first node based on the read conflict response being a highest priority non-data response that is received by the first node;
providing the requested data to the first node from a third node in response to the source broadcast request from the first node; and
placing the data provided by the third node in a cache associated with the first node based on the state of the conflict state machine.

35. (Original) The method of claim 34, wherein providing a source broadcast request from the first node comprises providing a source broadcast read request from the first node, the method further comprising:
providing a read conflict response from the first node to the second node in response to the pending broadcast read request of the second node, the read conflict response from the first node indicating that the pending broadcast read request of the second node conflicts with the source broadcast read request provided by the first node;
providing the requested data to the second node from the third node in response to the pending broadcast read request of the second node; and
placing the data provided to the second node by the third node in a cache associated with the second node.

36. (Original) The method of claim 34, wherein providing a broadcast request from the first node comprises providing a broadcast write request from the first node, the method further comprising:

providing a second conflict response from the first node to the second node in response to the pending broadcast read request of the second node, the second conflict response from the first node indicating that the pending broadcast read request of the second node conflicts with the broadcast write request provided by the first node; and

preventing placement of the data in a cache associated with the second node in response to the second conflict response provided by the first node.

37. (Original) The method of claim 36, further comprising reissuing the source broadcast read request of the second node as a forward progress protocol read request for the data from the second node in response to the second conflict response provided by the first node

38. (Currently Amended) A computer system comprising a hybrid cache coherency protocol that employs source broadcast protocol mode and a forward progress protocol mode, the computer system being operative to fill a cache line associated with a source node with requested data provided in response to a source broadcast protocol mode request for the data when there is a source broadcast protocol read conflict with another node in the computer system, the computer system being further operative to reissue a request for the data from a the source node using a forward progress protocol mode request for the data when there is a source broadcast protocol second conflict with another node in the computer system, the source broadcast second conflict having a higher priority than the source broadcast read conflict such that the computer system is operative to reissue a request for the data from the source node using the forward progress protocol mode request for the data when there is both a source broadcast protocol read conflict and a source broadcast protocol second conflict with one or more other nodes in the computer system.